



**Ashesi University**

**DEVELOPMENT OF A BIOMETRIC AUTHENTICATION VOTING  
SYSTEM FOR SENIOR HIGH SCHOOLS IN GHANA**

**THESIS**

**B.Sc. Computer Engineering**

**Emmanuel Yaw Manu Annor**

**2020**

**ASHESI UNIVERSITY**

**DEVELOPMENT OF A BIOMETRIC AUTHENTICATION VOTING  
SYSTEM FOR SENIOR HIGH SCHOOLS IN GHANA**

**CAPSTONE PROJECT**

Capstone Project submitted to the Department of Engineering, Ashesi  
University in partial fulfilment of the requirements for the award of Bachelor  
Of Science degree in Computer Engineering.

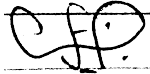
**Emmanuel Yaw Manu Annor**

**2020**

## DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:



.....

Candidate's Name:

Emmanuel Yaw Manu Annor

.....

Date:

May 29, 2020

.....

I hereby declare that the preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

## **Acknowledgement**

Sincere gratitude to all the people who advised and encouraged me to undertake this project. I could not have achieved this feat without your guidance and support.

I am grateful to my father Professor Serekye Yaw Annor and mother Victoria Adu-Owusuah Annor, for their love, prayers, care, and commitment shown to me throughout my undergraduate journey. Without your tremendous emotional support and investment, this journey would not have been a success.

To my able supervisor, Mr. David Amatey Sampah, God bless you a lot for your sacrifices and sleepless nights to review my works. Your guidance and suggestions through the design and implementation process have helped me to build more confidence as a Computer Engineering major to move further and work on several other projects.

My gratitude to Dr. Nathan Amanqauh, Acting Dean of Ashesi Engineering Department, for his further guidance, suggestions, and comments about this project. His immense contributions to this project, and others have impacted my life in my four-year study at Ashesi University.

## **Abstract**

Voting is practised in almost every Senior High School in Ghana to elect school leaders in various schools. In Ghana, several issues are facing senior high schools during voting processes. Some of these issues include: The voting process being costly, taking long hours to finish the voting process, Issues of election fraud (election manipulation or vote rigging), and generally the voting process being tedious, not secured, and trusted. It is required for every voting process to be transparent and able to withstand a variety of fraudulent behaviours. This project presents the design of a highly secured, cost-effective voting system designed with Flask python web framework running on a raspberry pi 3B+ (server). This system includes a fingerprint authentication as a form of security to prevent fraudulent behaviours. The results from the voting are displayed on an admin page, together with some other voting statistics.

**Key Words:** biometric authentication, Flask, web server, election fraud, cost-effective.

## Table of Content

Contents	
<b>DECLARATION</b>	i
<b>Acknowledgement</b>	ii
<b>Abstract</b>	iii
<b>Table of Content</b>	iv
<b>List of Tables</b>	vi
<b>List of Figures</b>	vii
<b>Chapter One</b>	1
<b>1.0 Introduction</b>	1
1.1 Introduction and Background	1
1.2 Problem identification	2
1.3 Objectives	3
1.4 Motivation and Justification	3
1.5 Expected Outcome	4
<b>Chapter Two</b>	5
<b>2.0 Literature Review</b>	5
2.1 Available Solutions	5
2.2 Reviewed Articles	6
<b>Chapter Three</b>	9
<b>3.0 Requirements and Methodology</b>	9
3.1 Thesis Design Objective	9
3.2 Design Decisions	9
3.3 System Requirements	10
<b>3.3.1 Functional Requirements</b>	10
<b>3.3.2 Minimum Requirement Specification</b>	11
<b>3.3.3 Non-functional Requirements</b>	11
<b>3.3.4 Use Case Scenario</b>	12
<b>3.3.5 Flowchart</b>	13
3.4 Architectural Design	14
<b>3.4.1 Software and Hardware implementation</b>	15
<b>Chapter Four</b>	19
<b>4.0 Implementation</b>	19
4.1 Experimental Setup	19
<b>4.1.1 Hardware unit development</b>	19

<b>Chapter Five</b> .....	23
<b>5.0 Results</b> .....	23
5.1 Web Application .....	23
5.2 Database .....	26
<b>Chapter Six</b> .....	28
<b>6.0 Conclusion</b> .....	28
6.1 Discussions.....	28
6.2 Limitations .....	28
6.3 Future works.....	29
<b>References</b> .....	30
<b>Appendix</b> .....	33

## **List of Tables**

Table 1: Key for Pugh Matrix .....	9
Table 2: Pugh Matrix for Microcontrollers.....	10
Table 3: Hardware and Software components description .....	17



## List of Figures

Figure 1: System Architecture .....	4
Figure 2: A student of Christ the King International High School using an electronic voting system ...	6
Figure 3: Use Case of the System .....	12
Figure 4: Flowchart of the System.....	13
Figure 5: The architectural design of the system .....	14
Figure 6: Schematic diagram for raspberry pi connection .....	15
Figure 7: Block diagram of software implementation .....	16
Figure 8: Inserting an SD Card into the Raspberry pi.....	19
Figure 9: Connecting the fingerprint module to the raspberry pi via PL2303 USB TTL Cable FTDI.	20
Figure 10: Connecting a raspberry pi to a 5" 800x480 TFT HDMI display .....	20
Figure 11: Connecting keyboard and mouse to the raspberry pi.....	21
Figure 12: Adding a public GPG (privacy guard) key to verify packages of code OSS that would be installed.....	22
Figure 13: Code for installing Code OSS IDE.....	22
Figure 14: Accuracy score and hash of a fingerprint during the first login test.....	23
Figure 15: Accuracy score and hash of a fingerprint during the second login test .....	23
Figure 16: Accuracy score and hash of a fingerprint during the third login test.....	23
Figure 17: Accuracy score when a user is not found in the database during login .....	23
Figure 18: An image showing currently available voting polls of the system. ....	24
Figure 19: Diagram shows available candidates in the Senior School Prefect poll. ....	25
Figure 20: View of the School Prefect poll results page after voting. ....	25
Figure 21: The table shows the candidate table created in the database. ....	26
Figure 22: The table shows the questions table created in the database. ....	27
Figure 23: Flask web server running on Raspberry pi .....	34
Figure 24: Directory of the project .....	35

Figure 25: Image of various flask extensions and frameworks imported .....	36
Figure 26: Code implementation of the login system .....	37
Figure 27: Code implementation of the login system (continues) .....	37
Figure 28: Code implementation of the signup system.....	38
Figure 29: Code implementation of signup System(continues) .....	38
Figure 30 Code Implementation of creating a new poll (Backend script) .....	39
Figure 31:Code Implementation of adding a new candidate to the poll created (Backend script) .....	39
Figure 32: HTML implementation of adding a new candidate to the poll created (Frontend script) ...	40
Figure 33: HTML implementation of adding a new candidate to the poll created (Frontend script) continues .....	40
Figure 34: JavaScript implementation of adding a new candidate to the poll created (Frontend script) .....	41
Figure 35: JavaScript implementation of adding a new candidate to the poll created (Frontend script) continues .....	41
Figure 36: Update route gets information posted from add page and adds it to the database.....	42
Figure 37: Code implementation of the voting page (Backend Script).....	42
Figure 38: Figure 19 Code implementation of the voting page (Backend Script) continues .....	43
Figure 39: Creating database tables as classes in code OSS IDE .....	43
Figure 40: Creating Flask Forms as class .....	44

# **Chapter One**

## **1.0 Introduction**

### **1.1 Introduction and Background**

Under the Constitution of the Republic of Ghana (Amendment) Act, 1996 Act 527, article 42; Every citizen of Ghana of eighteen years of age or above and of sound mind has the right to vote and is entitled to be registered as a voter for public elections and referenda [1]. According to [2], the Government of Ghana was created as a parliamentary democracy, followed by alternating military and civilian governments. In January 1993, the military government gave way to the Fourth Republic after presidential and parliamentary elections in late 1992. Since then, Ghana has been enjoying a peaceful and fair democracy. During these times, elections were conducted using the traditional system of voting (ballot paper voting). In 2012 the Electoral Commission of Ghana proposed and adopted a new biometric system of voting.

In Ghana, fingerprint verification was used to verify voters by the Electoral Commission before they cast their votes during the 2012 elections. In the face of advancing ICT trends and emerging challenges in manual voter registration, the Electoral Commission of Ghana took the bold decision to replace the current voters' register with a finger biometric register for subsequent elections [3]. Also, some major issues they were averting included [4], preventing multiple voting, preventing voter impersonation, and preventing ballot stuffing.

In every senior high school (SHS) in Ghana, new student leadership is selected at the beginning of a new academic year to forward the interests of students to the schools' administration. When a government is elected, they must plan and organize all activities in the school. Since Ghana is a democratic country, most schools choose student leaders through a voting process. Every student in the school is encouraged to be patriotic and participate in the school's general

elections to elect a new student leadership because every voice matter and needs to be heard during this process.

## **1.2 Problem identification**

An election system must be sufficiently robust to withstand a variety of fraudulent behaviours and must be sufficiently transparent and comprehensible that voters and candidates can accept the result of the election [5]. The researcher conducted ethnographic research by visiting three different senior high schools in Kumasi, Ashanti Region of Ghana, on January 7, 2020. The following are some problems stated by students and staffs of various schools after the research was conducted:

1. The voting process is costly: the cost of voting manually mostly includes money for buying and printing ballot papers and funds to provide incentives to staff to help the election process to be a success.
2. Taking longer hours to finish the voting process hence sometimes classes needs to be cancelled throughout the day.
3. Most students do not trust the voting process because of issues of human errors (counting mistakes, etc.).
4. Issues of election fraud (election manipulation or vote rigging).
5. The process is generally tedious, not secured, and trusted.

According to the Head of Science Education and ICT Unit under the Secondary Education Division of the Ghana Education Service [6], there is a need to develop, implement and popularize science and technology through various activities in Ghana. All schools in Ghana are greatly encouraged to teach and practicalize Information Technology (IT) education to their students because of its wide range of advantages. In this light, it is only beneficial an efficient

system is proposed and implemented in various senior high schools to make voting safe, fair, and transparent.

### **1.3 Objectives**

The aims of carrying out this project are:

1. Designing a hardware implementation to help students in senior high schools vote with ease.
2. Exploring the possibilities of a highly secured voting system to make elections safe, fair, and transparent.

### **1.4 Motivation and Justification**

It is about time students in Ghanaian based colleges took the responsibility of identifying problems around them and solving them. This would give students and graduands the experience to compete in the world of work since they would already have skills of creating and implementing successful projects; in the long term, this will create a positive impact in Ghanaian tertiary institutions.

Also, due to the evolution of technology in the world, it is only feasible to excite teenagers (mainly senior high school students) with such projects to spur their interest in information technology. The use of ICT in the learning environment can bring about a rapid change in the student 's performances [7]. Most students will be intrigued and challenged to take their studies seriously so they can excel and build such projects on their own [7]. In the long run, this project can have a positive impact on the nation and urge more of the youths to focus on the technology industry.

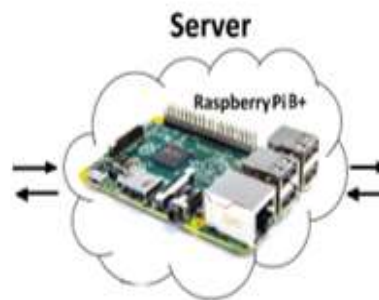
## 1.5 Expected Outcome

This project's implementation is to: design a webpage connected to a fingerprint sensor(scanner), which would be connected to a microcontroller (Raspberry pi 3 B+) to help students vote in their schools. This implementation will allow students to be able to login with their credentials (Student name, Student password, and fingerprint ID) before voting. A system like this must be highly secured. Therefore, considerations such as a multi-factored authentication (password and fingerprint) and other forms of security procedures will be factored into the development. Also, to make sure the election is safe and fair, every student can only vote once since they all have unique fingerprints, which would be checked in a database. The process must be as efficient and fast as possible.

PC Client



Device



*Figure 1: System Architecture*

This system's architecture follows the client-server archetype; that is, a client (in this case execution from the raspberry pi's browser) will make a request to the server; hence the server will reply to the client. The server in the raspberry pi is responsible for managing the system (including the fingerprint sensor attached to the pi).

## **Chapter Two**

### **2.0 Literature Review**

#### **2.1 Available Solutions**

Some senior high schools in Ghana have already started the implementation of electronic voting in their schools. Such schools include Christ the King International High School, Mawuko Girls SHS, and Nyankumasi Ahenkro S. H. S. and others. These schools use a somewhat similar system: a combination of basic HTML (HyperText Markup Language) standard markup language for creating web pages and PHP (Hypertext Pre-processor), a programming language for web developers. Students select various aspiring candidates on the webpage, which calculates the results of the election after the election process ends. Various reasons for the implementation of these systems include:

1. [8] Taking out the cumbersome process of sorting out and counting ballots after voting.
2. [9] Making students aware of the various application in the Information Communication Technology space and preparing them for electronic voting systems, which would soon dominate primary elections across nations.
3. [10] For students to be able to vote for preferred candidates in less than a minute.



*Figure 2: A student of Christ the King International High School using an electronic voting system*

The biometric authentication factor added to the proposed method in this paper will make it more secure than the above-discussed systems because it uses multi-factor authentication and a fingerprint scanner, which will improve the security and authentication of the system. This form of security (biometric) is used across significant elections in various nations.

## **2.2 Reviewed Articles**

1. [11] Electronic voting has been successfully implemented in countries like Australia, Belgium, Estonia, France, and many others. Since the introduction of this voting system, the various countries have benefited, and the system has helped solve most problems associated with manual voting. The benefits Australia enjoys by implementing electronic voting are; provision of a secret ballot for blind and low-vision



voters, more natural delivery of remote voting services, and secure ballot-handling. From this inspiration, a well-thought-through implementation of an electronic voting system can solve the problem identified in senior high schools in Ghana.

2. A Journal by Himanshu Vinod Purandare, Akash Ramswaroop Saini and others on a proposed system: Online Voting System that Utilizes a Facial Recognition Resource

[12] This voting system uses a facial recognition resource to secure the system by allowing only authentic voters to have access to it. Also, it uses a One Time Password (OTP) to provide extra security to the system. A database was implemented using XAMPP server; hence voters' details can be retrieved whenever needed. This application can run on all android phones, but most android phones do not use the facial recognition software; hence such users cannot use this application.

3. A Journal by Jagdish B. Chakole and P. R. Pardhi on a proposed system: Internet Voting System Designed for Corporate Elections

[13] To keep the system very secured (providing security to casted votes). This is done by preventing active (tampering casted vote) and passive intruders (access cast votes and make changes) from having access to it. To achieve this, the concept of cryptography and digital signatures were used. Hence votes will be encrypted during voting and decrypted for votes calculations.

4. A Journal by Vinayak Bharadi and Dhvani Shah on a proposed system: IoT Based Biometrics Implementation on Raspberry Pi

[14] This implementation introduces a secured technology based on information security, shedding more light on how biometrics can leverage cloud's vast computational resources and striking properties of flexibility, scalability, and cost

reduction to reduce the cost of the biometrics system requirements of different computational resources (i.e., processing power or data storage) and to enhance the performance of biometrics systems' processes (i.e., biometric matching). This is done using a Raspberry Pi to build a low-cost biometric system.

## Chapter Three

### 3.0 Requirements and Methodology

#### 3.1 Thesis Design Objective

- i. To design a hardware implementation that will help students in senior high schools vote with ease.
- ii. To explore the possibilities of a highly secured voting system that will make elections safe, fair, and transparent.
- iii. To leverage the cloud's vast computational resources (storage, flexibility, cost efficiency), which would reduce the cost of biometric system requirements.

#### 3.2 Design Decisions

[15] To make the best decision and choose between the alternatives available that will contribute to the development of the voting system, a Pugh matrix and insights from the literature were considered. A Pugh matrix is a decision-making model designed to choose and compare available alternatives using its criteria's:

*Table 1: Key for Pugh Matrix*

Key	Meaning
S	Same or Neutral
+	Better than
-	Worse than
0	Baseline

*Table 2: Pugh Matrix for Microcontrollers*

		Baseline	Weight	Node MCU	Atmega8a	FRDM-kl25z
Criteria		Raspberry pi 3B+				
Cost		0	2	+1	+1	+1
Power Consumption		0	4	+1	+1	+1
Processing Power		0	4	-1	-1	-1
Memory segments		0	3	-1	-1	-1
Total Plus (p)			14	6	6	6
Total Minus (m)			0	7	7	7
Total Neutral			0	0	0	0
Total (p + m)			14	-1	-1	-1

From the table above (Table 2), the Raspberry pi 3B+ is the best overall alternative in terms of cost, relatively higher processing power, power consumption, and memory segments.

### 3.3 System Requirements

#### 3.3.1 Functional Requirements

The functional requirements of the system state precisely what the system will do; this includes:

- i. System signup must work successfully.
- ii. The system login must work successfully.
- iii. The system should be able to send user data into the database
- iv. The system must detect the fingerprints of users.
- v. The system should be able to calculate the total vote.

### 3.3.2 Minimum Requirement Specification

The minimum requirements specification needed for the system to function correctly and efficiently includes:

- i. Raspberry pi 3B+
- ii. Memory card for raspberry pi 3 B+ 2GB
- iii. Web browser
- iv. 5" 800x480 TFT HDMI display or LCD Screen
- v. Fingerprint scanner

### 3.3.3 Non-functional Requirements

#### 3.3.3.1 *Software Quality Attributes*

To describe various criteria that judge the final requirements

- i. Performance: The system should be user-friendly so, it accomplishes its various tasks effectively; signing up, logging in, voting for a candidate, displaying results after voting, and logging out.
- ii. Speed: Publishing data (results) from the database to the system should be fast.
- iii. Accuracy: The system will compute its calculations correctly and act accordingly.
- iv. Timing: The system should be responsive 24 hours a day during and after the voting process. Except it is undergoing maintenance and upgrades.
- v. Security: The system should be protected from unauthorized users. Also, the system should allow its users to vote **once**.

### 3.3.3.2 Safety Requirements

The following will be considered to provide the best level of safety in the system:

- i. The system includes a biometric authentication to provide security to its users before permitting login and vote.
- ii. The system should hash voter's information when the vote is cast to prevent votes being traced.
- iii. The system should provide a means to log out of the system.

### 3.3.4 Use Case Scenario

The following explains the interactions between an admin, users, and the voting system.

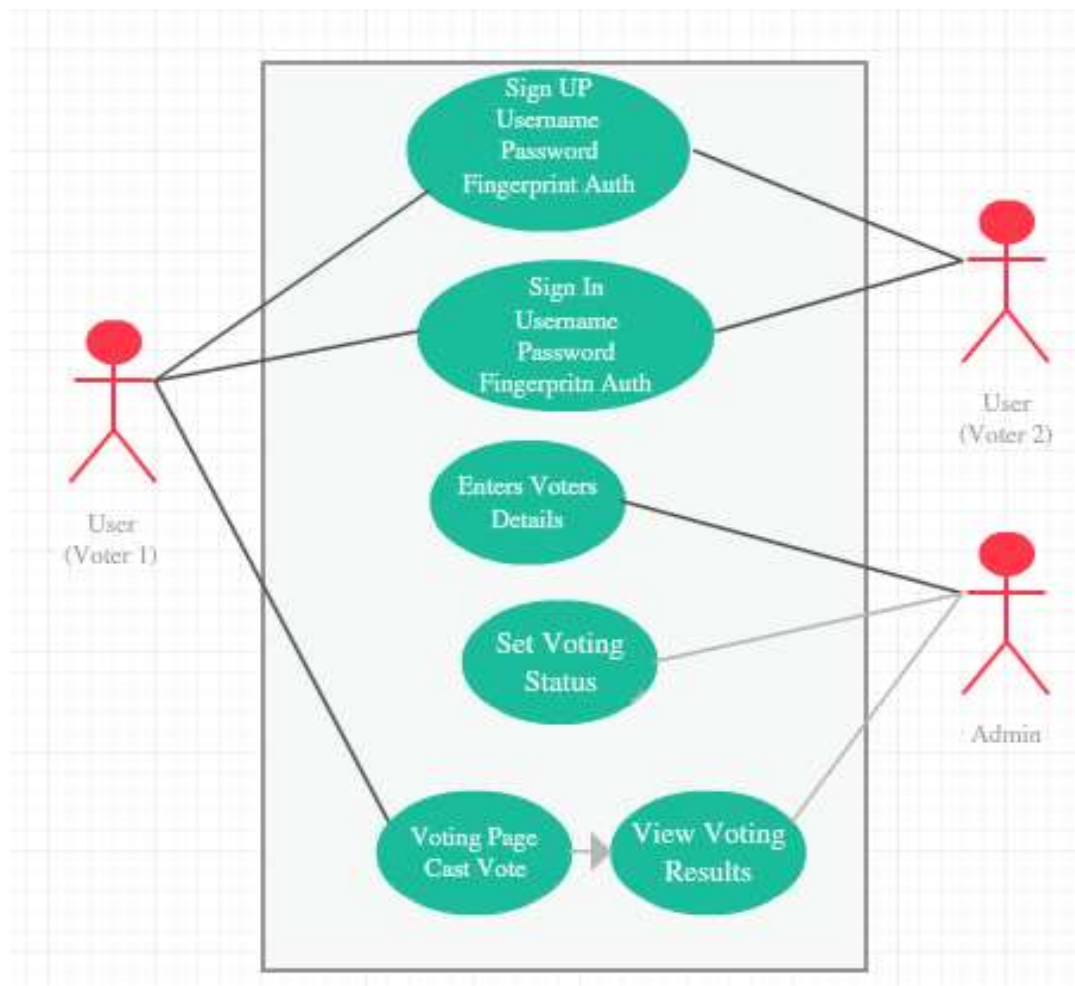


Figure 3: Use Case of the System

### 3.3.5 Flowchart

The flow chart of the system is illustrated using the diagram below

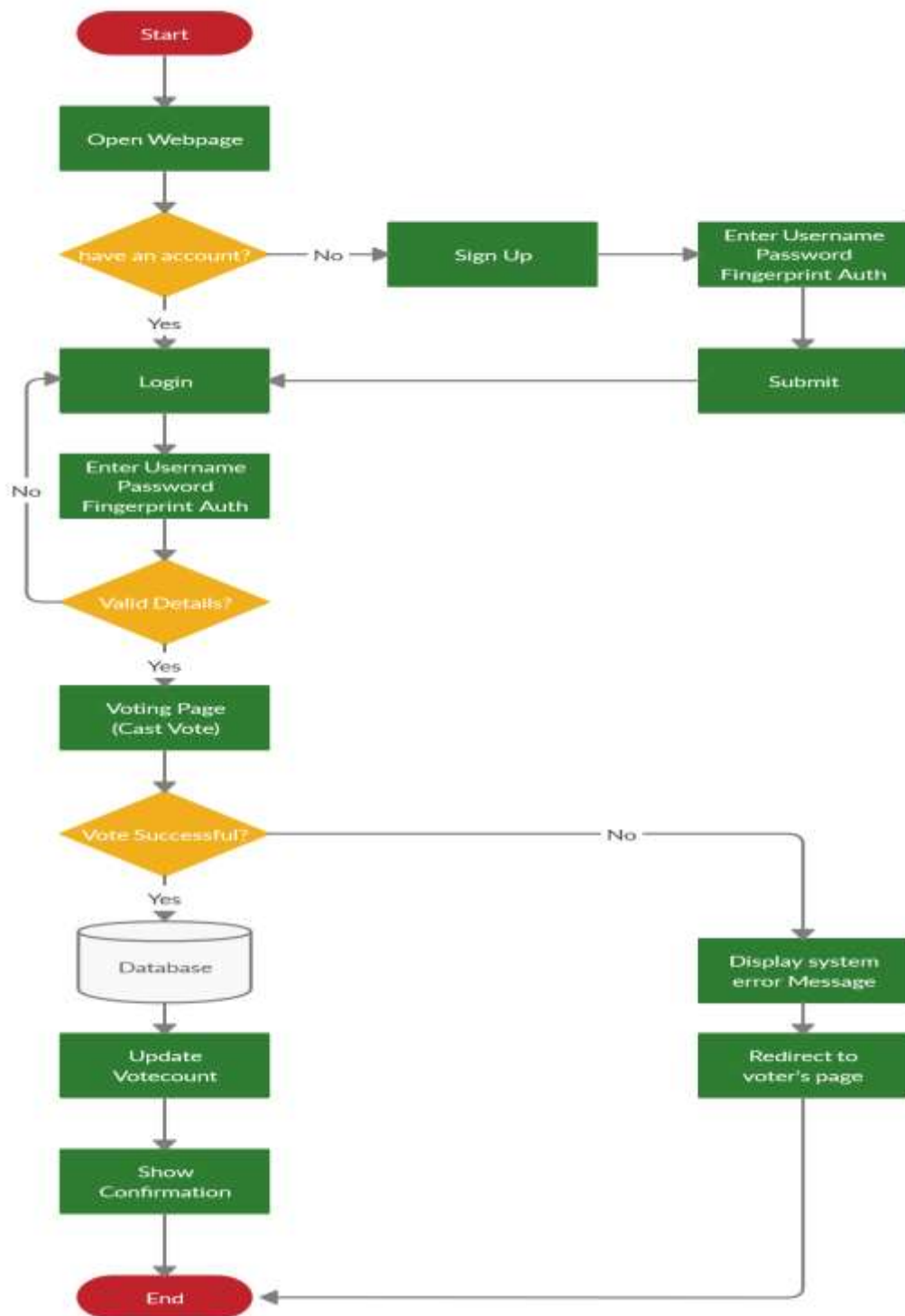
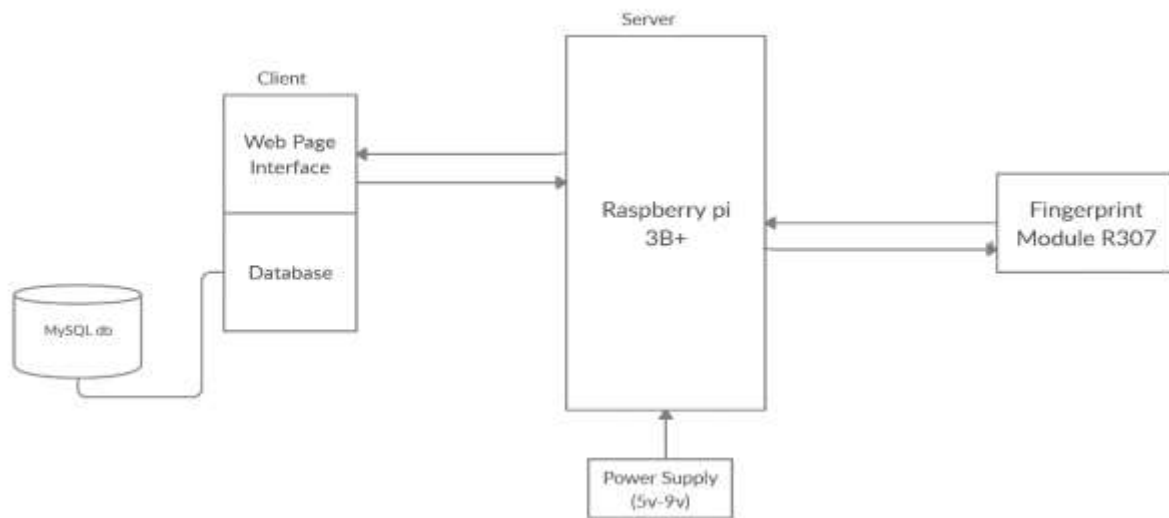


Figure 4: Flowchart of the System

### 3.4 Architectural Design

The architectural designs of the system explain the main implementation and relationships of both software and hardware in the system. The figure below shows the architectural design of the system:



*Figure 5: The architectural design of the system*



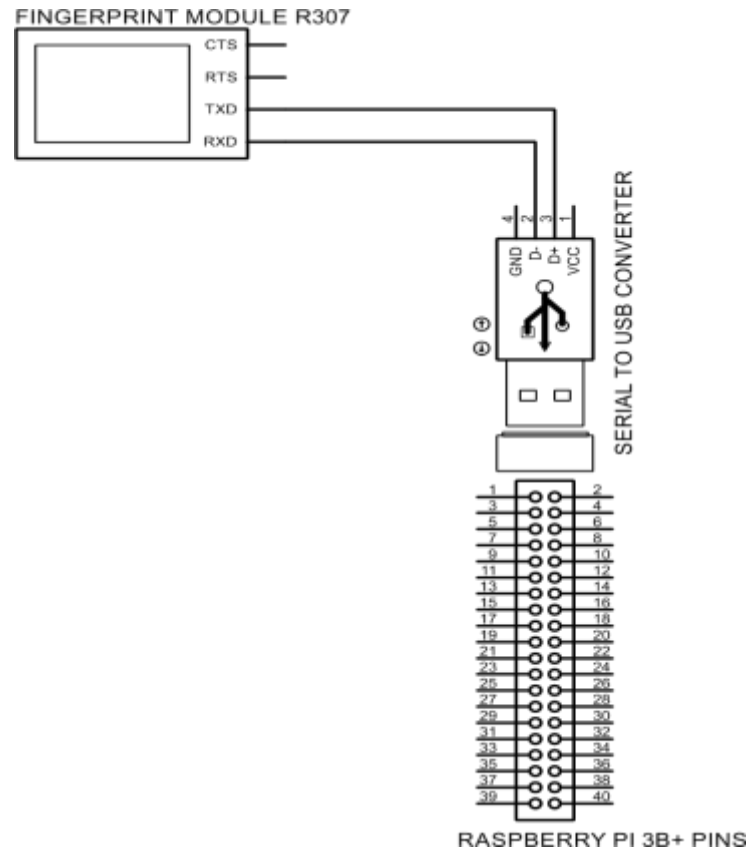
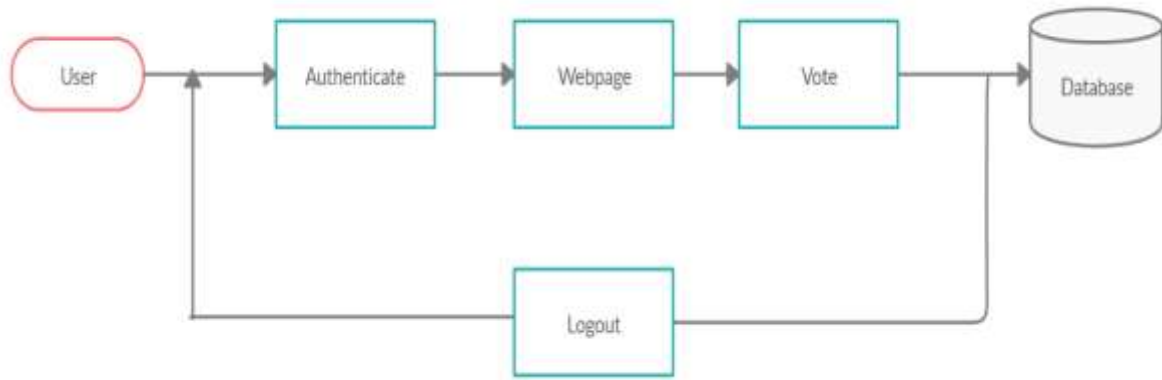


Figure 6: Schematic diagram for raspberry pi connection

### 3.4.1 Software and Hardware implementation

Figure 6 explains how the various components in the system will be connected. The fingerprint sensor R307 will be connected to the microcontroller (raspberry pi 3B+) via serial to USB converter (as shown in the schematic diagram in figure 4). With the raspberry pi acting as a server, users will interact (vote and view voting results) with the system via the webpage interface displaying on a 5" 800x480 TFT HDMI display screen or LCD Screen. The webpage is connected to MySQL database, which stores all user details and voting statistics. The entire system is powered by an output voltage between 5v-9v.



*Figure 7: Block diagram of software implementation*

#### **3.4.1.1 Software unit implementation**

This section is categorized into three subsections: building the python server using Flask, web development with HTML5, CSS, Bootstrap, jQuery, and MySQL database and web deployment with Heroku.

- a) To create a simple server-side application that is easily compatible with fingerprint sensor reader R307 library, does not consume a lot of memory, can easily accommodate changes, has better testability, and better performance, Flask was chosen. Flask is a micro web framework server gateway interface written in python. To efficiently and securely build the server-side application, various Flask extensions including Flask-login to create a user session management for Flask, Flask-WTForms to provide an interactive user interface for the login and signup pages, Flask-SQLAlchemy to offer a secure interaction and integration with MySQL database using SQLAlchemy, Flask-Werkzeug security to provide security interface for user data, were used to form the skeleton of the web application. These different extensions and frameworks form wrappers around the Flask framework to facilitate the request and response process. Visual studio code-OSS was the preferred IDE used since there is no official release of visual studio code for ARM devices like the Raspberry pi.

- b) [16] To design a responsive, interactive and user-friendly views, client-side frameworks, sheet languages and other open-source frameworks such as HTML5: the standard mark-up language for document rendering in browsers, Bootstrap: a stylesheet language framework used to control the appearance of HTML documents and jQuery: a JavaScript library used to handle events, and Ajax (gives access to asynchronous web applications.) was used.
- c) The app will be hosted and deployed on Heroku, a platform as a service (PaaS) application that is compatible with python web framework and MySQL database. It has been in existence for a long time; hence it has a large community that's ready to offer help if needed.

The description of components in the system, how they relate to each other is further explained in the table below. Pictures of the hardware components can be found in appendix A.

*Table 3: Hardware and Software components description*

Device	Application
Raspberry pi 3B+	[17] A powerful microcontroller (small single-board computer) that has a 1.4GHz 64-bit quad-core processor and 1 GB RAM. It also has a LAN wireless and Bluetooth connectivity to create ideal solutions. In this situation, it is used to host a full website.
5" 800x480 TFT HDMI	[18] This is a mini LCD/HDMI display with a high picture resolution, fast response time, and full-colour display. It will serve as a display for the raspberry pi.

Fingerprint sensor R307	The fingerprint sensor would be used as a means of authentication for users before they can vote. It has an optical green backlight and is USB2.0/UART compatible.
Python Flask	[19] Flask is a micro web framework server gateway interface written in python that makes it easier to scale up to complex applications. To efficiently and securely build the server-side application on the raspberry pi, various Flask extensions be used.
MySQL database	A relational database system that would be used to store user data and voting statistics.
PL2303 USB TTL Cable FTDI	A serial to USB converter that connects the fingerprint sensor to the raspberry pi
Bootstrap	An open-source framework that makes it easier to design web interfaces
Heroku	Web hosting platform

## Chapter Four

### 4.0 Implementation

#### 4.1 Experimental Setup

In developing the voting system, the various parts of the system were modularized into different units, mainly the software and hardware unit. The software unit encompasses building the server, web development, and deployment. The hardware unit comprises the interconnection of all the physical hardware components used in the system. The two units precede a system testing phase.

##### 4.1.1 Hardware unit development

The various components that were used in the development of this system to achieve different functionalities, include raspberry pi 3B+ microcontroller which will serve as the server for the web application, 5" 800x480 TFT HDMI display which will serve as the display for the microcontrollers' operating system (main user interface), Fingerprint Sensor Reader R307 to read the fingerprint of users, 16GB flash memory card that serves as a storage device for the microcontroller, a keyboard and mouse to serve as primary input devices for the raspberry pi microcontroller and a PL2303 USB TTL Cable FTDI which will serve as a gateway connecting the fingerprint sensor to the raspberry pi microcontroller. The following steps explain how all the devices were systematically connected (Hardware unit implementation):

1. Insert a 16GB flash memory with a Raspbian operating system installed into the Raspberry pi 3B+ memory slot.



*Figure 8: Inserting an SD Card into the Raspberry pi*

2. Connect one side of the PL2303 USB TTL Cable FTDI (Serial to USB converter) to the USB port of the Raspberry pi 3B+ and the other side to the Fingerprint Sensor Reader R307.

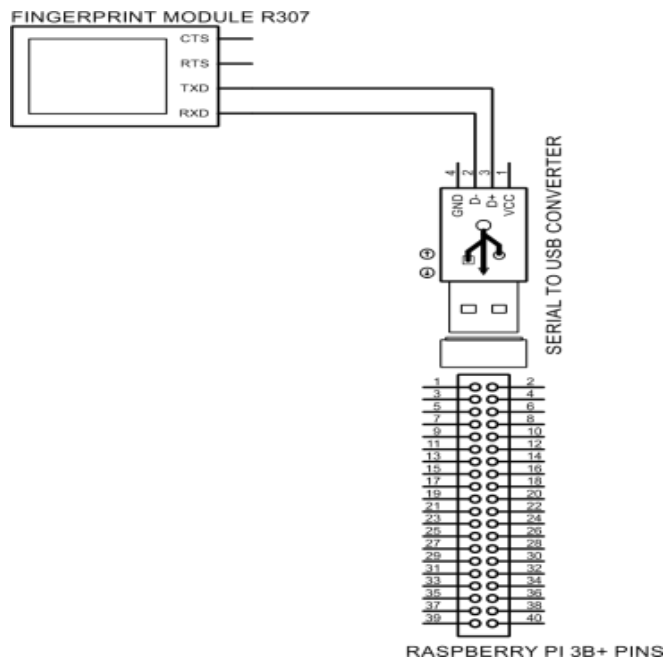


Figure 9: Connecting the fingerprint module to the raspberry pi via PL2303 USB TTL Cable FTDI

3. Connect the 5" 800x480 TFT HDMI display to the HDMI port of the Raspberry pi 3B+.

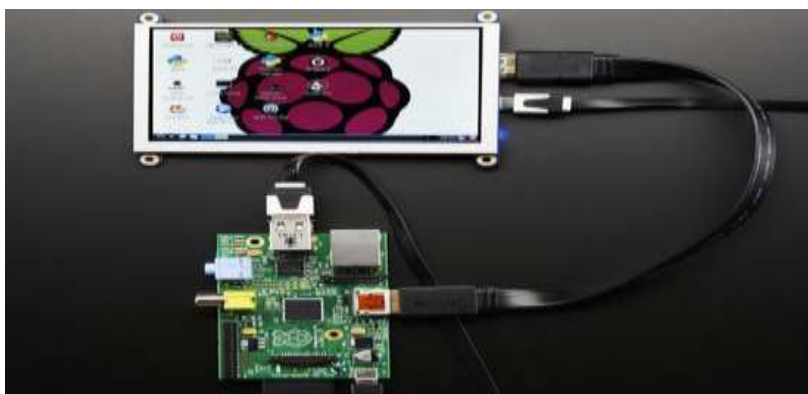


Figure 10: Connecting a raspberry pi to a 5" 800x480 TFT HDMI display

4. Connect a keyboard and mouse to the raspberry pi 3B+ via USB ports.



*Figure 11: Connecting keyboard and mouse to the raspberry pi.*

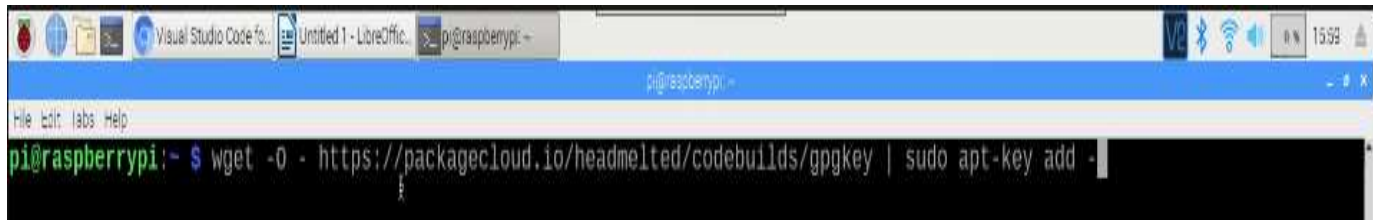
5. Switch on the power supply of the Raspberry pi 3B+ to turn it on.

#### **4.1.2 Software unit development**

The complete web application has six main interfaces: The **Signup page**: provides users with fingerprint authentication and signup option together with a username and a password. **Login page**: gives access to users to log in to the system and have access to remaining views. **Create Poll page**: provides an administrator with an option to create a voting poll(questions). **Available Polls**: users view available questions or voting polls so they can select and vote accordingly. **The voting page** gives users an option to vote for a candidate once. **Results page**: display the results of the ongoing voting to administrators and users after they vote.

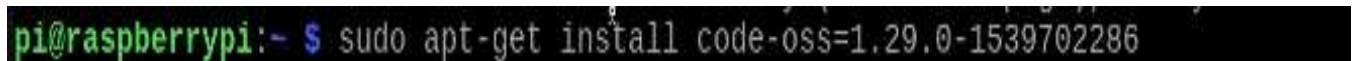
Before building these pages in FLASK:

1. Download Visual studio code-OSS preferred IDE for this project on the raspberry pi 3B+.
  - a. The first thing to do is to install a privacy guard key to decrypt and encrypt Code OSS packages that would be installed.



*Figure 12: Adding a public GPG (privacy guard) key to verify packages of code OSS that would be installed*

- b. Install Code OSS with the following syntax in the terminal



*Figure 13: Code for installing Code OSS IDE*

2. Various files and folders in Appendix C were created
3. From the terminal, python three was installed; also, flask and all its frameworks and extensions mentioned were installed. After installation flask was initialized. Appendix B shows the FLASK server running (executing main.py as a python three file).
4. The main.py file contains all backend scripts used in the project, in this file, flask, all its extensions and framework (including Flask-SQLAlchemy for database purposes) were imported and initialized. Appendix D shows a visual representation of this.
5. Code for all the various pages (signup, login, create a poll, available polls, voting page, and result page) were written in main.py. Also, all HTML and CSS files for the frontend frameworks were written. This can be shown in Appendix E.
6. These processes come together to form the software unit of this system.

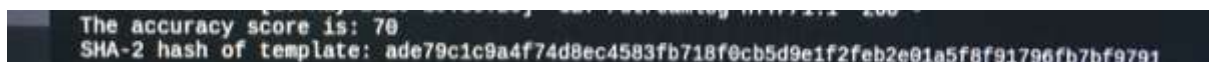


## Chapter Five

### 5.0 Results

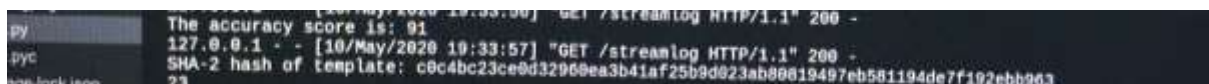
#### 5.1 Web Application

The web application is hosted on the Raspberry pi 3B+. Several voting tests were done on the system to ensure the accuracy and reliability of the application. In testing the signup and login part of the system: the focus was placed on the accuracy of the fingerprint sensor R307. The more accurate the fingerprint sensor works, the more reliable the data fetched from the database is. The following diagrams show the accuracy score from the fingerprint sensor and the user data stored in the database after signing up in the system.



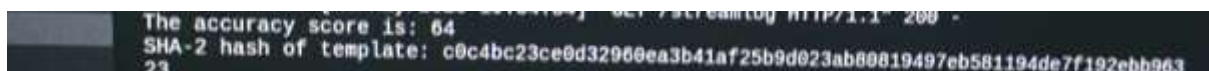
```
The accuracy score is: 70
SHA-2 hash of template: ade79c1c9a4f74d8ec4583fb718f9cb5d9e1f2feb2e01a5f8f91796fb7bf9791
```

*Figure 14: Accuracy score and hash of a fingerprint during the first login test.*



```
The accuracy score is: 91
127.0.0.1 - - [10/May/2020 19:33:57] "GET /streamlog HTTP/1.1" 200 -
SHA-2 hash of template: c0c4bc23ce0d32960ea3b41af25b9d023ab00819497eb581194de7f192ebb963
```

*Figure 15: Accuracy score and hash of a fingerprint during the second login test*



```
The accuracy score is: 64
SHA-2 hash of template: c0c4bc23ce0d32960ea3b41af25b9d023ab00819497eb581194de7f192ebb963
```

*Figure 16: Accuracy score and hash of a fingerprint during the third login test*



```
The accuracy score is: -1
```

*Figure 17: Accuracy score when a user is not found in the database during login*

From the test conducted, the average accuracy score after three tests is 75%. Although the fingerprint sensor R307 might be constrained, it can be reliable in terms of reading fingerprints of users.

Before a user can cast a vote on the system, the user needs to select one of the available polls created by an administrator.



*Figure 18: An image showing currently available voting polls of the system.*

After selecting a poll, a user can vote for a candidate once. This voting is done by selecting one of the available candidates.

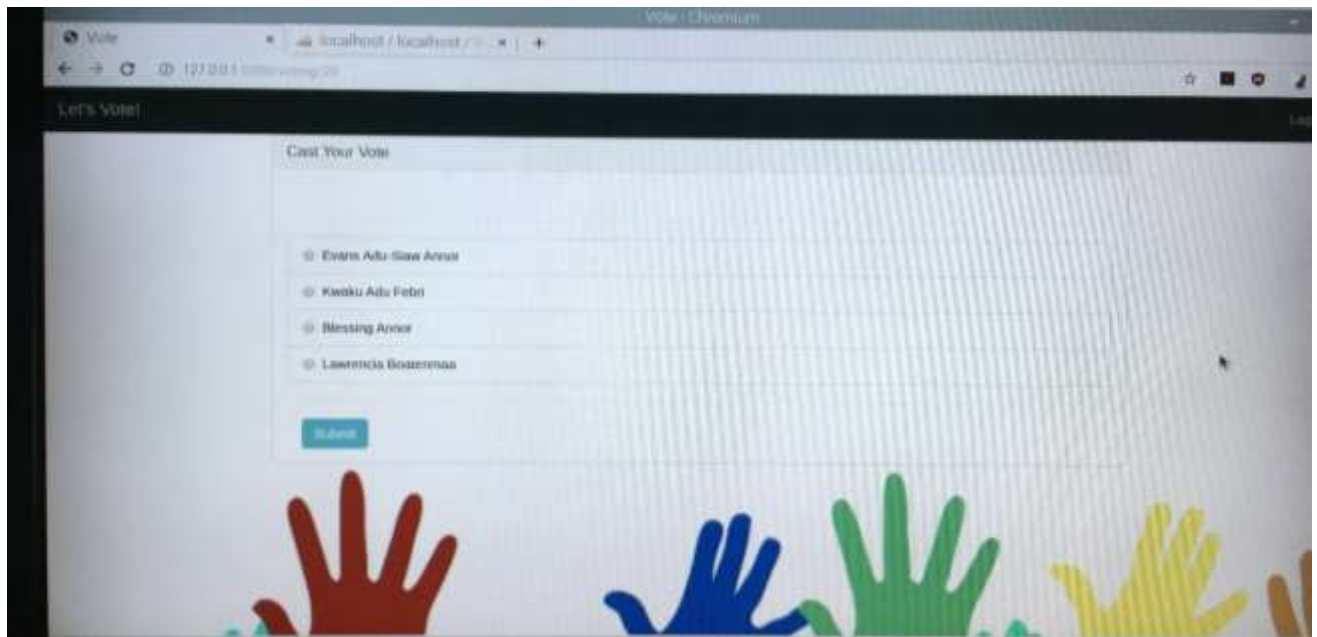


Figure 19: Diagram shows available candidates in the Senior School Prefect poll.

After voting for a candidate, a user can view the results page of the available polls.

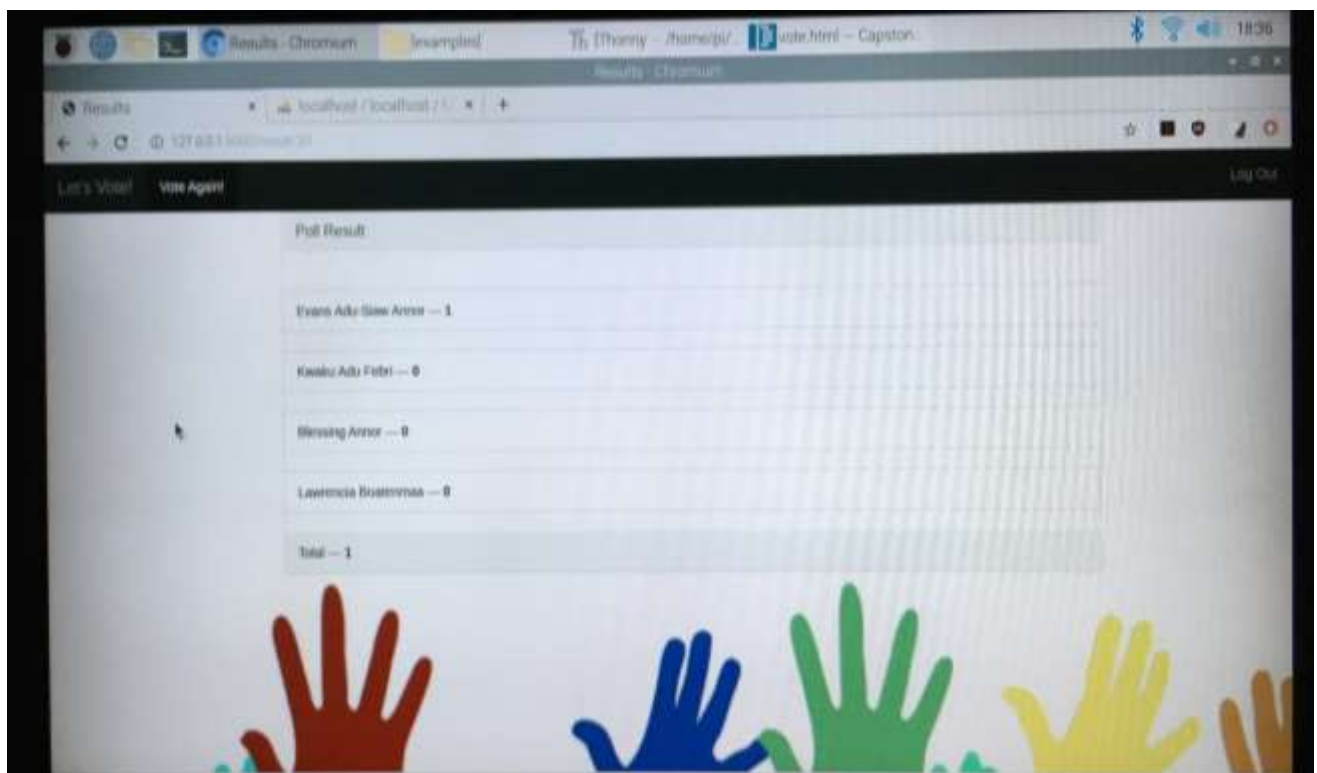


Figure 20: View of the School Prefect poll results page after voting.

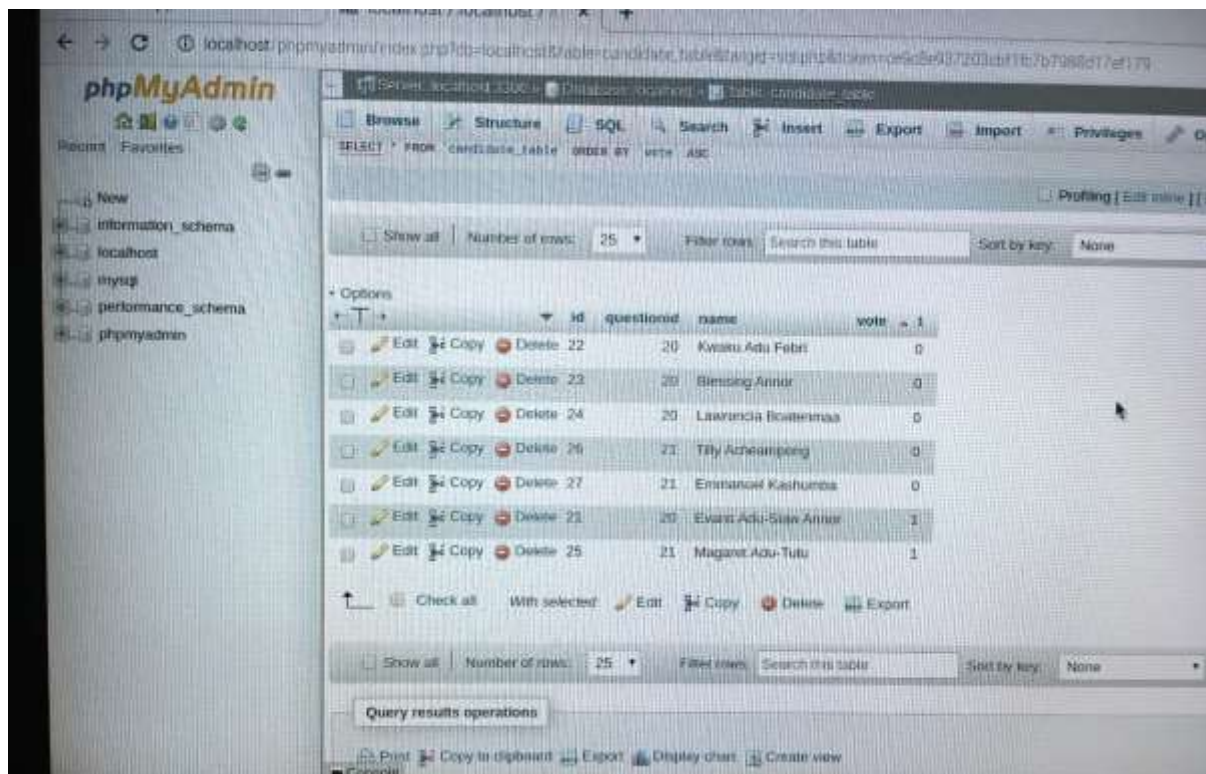
On average, it takes approximately about two minutes to signup, login, cast a vote, and view the results.

When the registration (signup) process is already completed, it takes approximately a minute to cast a

vote and view the results. This is possible due to how easy it is to navigate through the pages of the voting system.

## 5.2 Database

The database is implemented in a way to avoid multiple voting and to make voting anonymous. The following images show two tables created in the database. The question table (which shows descriptions of all polls) and the candidate table (which shows the available candidates in the system and their number of votes cast).



The screenshot shows the phpMyAdmin interface with the 'candidate' table selected. The table structure is as follows:

id	questionid	name	vote
22	20	Kwaku Adu Fobis	0
23	20	Bisming Annur	0
24	20	Lawrence Boateng	0
26	21	Tilly Acheampong	0
27	21	Emmanuel Kashumba	0
28	20	Evans Adu-Sark Annur	1
29	21	Margaret Adu-Tutu	1

Figure 21: The table shows the candidate table created in the database.

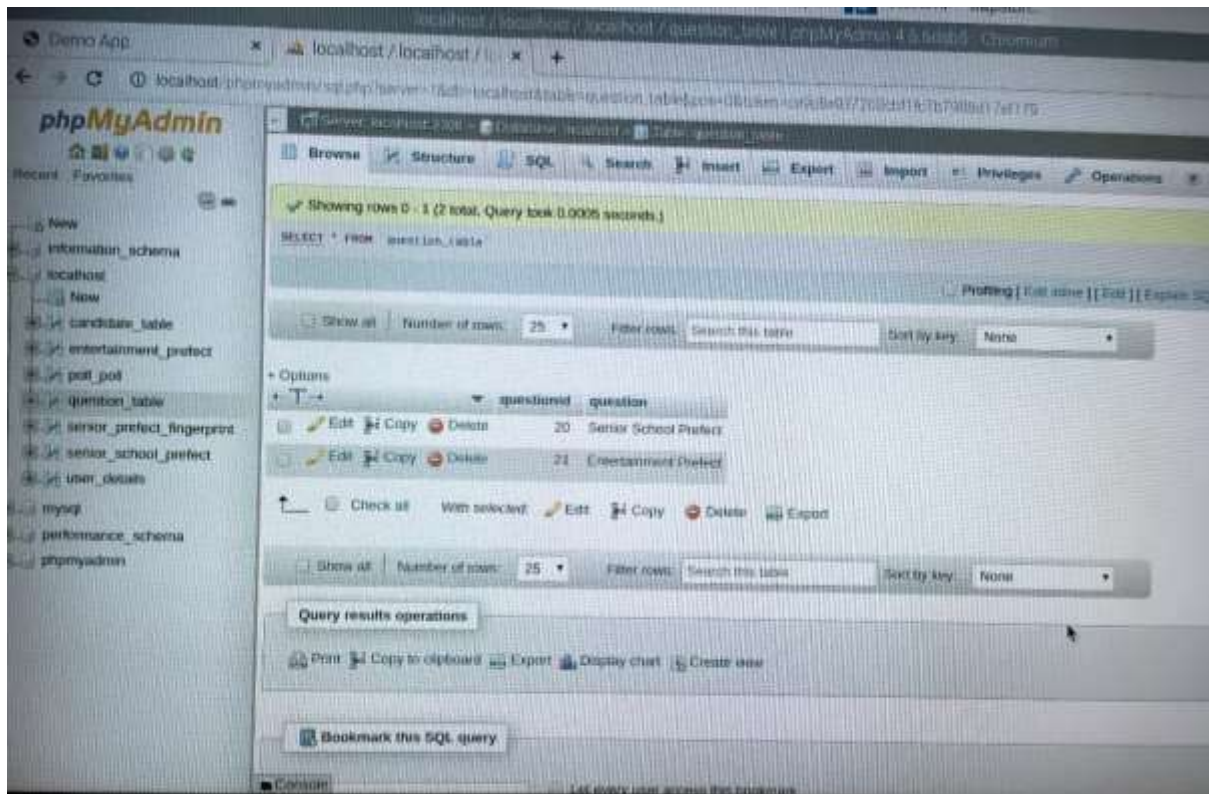


Figure 22: The table shows the questions table created in the database.

## **Chapter Six**

### **6.0 Conclusion**

#### **6.1 Discussions**

In this project, a secured, biometric authentication voting system has been presented. This system is safe, fair, and transparent; hence every user is obligated to vote once. The cost of the system is a onetime cost (recurrent cost), that is the cost to acquire all the components used to implement the system. This makes this system relatively cheaper as compared to the cost involved in the traditional voting system. All the equipment used in the project setup was carefully chosen and designed. The expected outcome of this project fulfils the objectives stated:

1. A hardware implementation to help students in senior high schools vote with ease.
2. Explores the possibilities of a highly secured voting system to make elections safe, fair, and transparent.

The result of the analysis proves that users can easily navigate and vote with the system within a shorter period.

#### **6.2 Limitations**

The current limitation of this project is that it is not hosted on an online platform yet. The focus was placed on developing a working system that fulfils all the requirement specifications stated.

### **6.3 Future works**

The current system is required to be hosted on an online platform to allow it to be active and live on the internet. This can improve the overall security of the website and make the application readily available to all end users.



## References

- [1] Constitution of the Republic of Ghana [Ghana], January 7 1993, available at:  
<https://www.refworld.org/docid/3ae6b5850.html> [accessed January 27 2020]
- [2] Thecommonwealth.org. (n.d.). Ghana: History | The Commonwealth. [online] Available at:  
<https://thecommonwealth.org/our-member-countries/ghana/history> [Accessed January 27 2020].
- [3] Bafo,A.2011Biometric Voter Registration: What You Should Know, Available at:  
[http://ghanamagazine.com/factual/politics/biometric-voter-registration-what-you-should-know/#.VEqTQPl\\_sTI](http://ghanamagazine.com/factual/politics/biometric-voter-registration-what-you-should-know/#.VEqTQPl_sTI) [Accessed: Accessed January 17 2020]
- [4] Aceproject.org. (2012). *Biometric voter verification in Ghana* —. [online] Available at:  
<http://aceproject.org/electoral-advice/archive/questions/replies/382803236> [Accessed 27 Jan. 2020].
- [5] K. Mythil, K. Kanagavalli and B. Shibi, *Ijcsmc.com*, 2014. [Online]. Available:  
<https://ijcsmc.com/docs/papers/February2014/V3I2201499a62.pdf>. [Accessed: 28-Jan- 2020].
- [6] Ges.gov.gh. (n.d). Secondary Education Division. [online] Available at:  
<https://ges.gov.gh/2019/07/15/secondary-education-division/> [Accessed 17 Jan. 2020].
- [7] Fathima, S. (2013). Challenges of ICT in teaching learning process. *International Journal of Engineering And Science*, 2, 51-54.
- [8] Nyavor, G. (2019). Christ, the King Int'l School, breaks ground with electronic voting system. Retrieved 17 January 2020, from  
<https://www.myjoyonline.com/news/2019/April-2nd/christ-the-king-intl-school-breaks-ground-with-electronic-voting-system.php>



- [9] N. Akpablie, "Mawuko Girls SHS introduces e-voting in school elections," *Citi 97.3 FM - Relevant Radio. Always*, 2018. [Online]. Available: <http://citifmonline.com/2018/02/mawuko-girls-shs-introduces-e-voting-school-elections/>. [Accessed: 17- Jan- 2020].
- [10] D. Kaku, "Nyankumasi Ahenkro S.H.S introduces e-voting in school election," *Ghanaweb.com*, 2018. [Online]. Available: <https://www.ghanaweb.com/GhanaHomePage/NewsArchive/Nyankumasi-Ahenkro-S-H-S-introduces-e-voting-in-school-election-636791>. [Accessed: 17- Jan- 2020].
- [11] Lundie, R. (2019). *Electronic voting at federal elections – Parliament of Australia*. [online] Aph.gov.au. Available at: [https://www.aph.gov.au/About\\_Parliament/Parliamentary\\_Departments/Parliamentary\\_Library/pubs/BriefingBook45p/ElectronicVoting](https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/BriefingBook45p/ElectronicVoting) [Accessed 13 Oct. 2019].
- [12] Purandare, H., Saini, A., Pereira, F., Mathew, B. and Patil, P. (2018). *Application For Online Voting System Using Android Device - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/8537284> [Accessed 9 Nov. 2019].
- [13] Chakole, J. and Pardhi, P. (2013). *The Design of Web-Based Secure Internet Voting System for Corporate Election* [online] semanticscholar.org. Available at: <https://www.semanticscholar.org/paper/The-Design-of-Web-Based-Secure-Internet-Voting-for-Chakole-Pardhi/2ccf0ca31ed727329177b263d432e2b34afe8b28> [Accessed 9 Nov. 2019].
- [14] D. Shah and V. Bharadi, "IoT Based Biometrics Implementation on Raspberry Pi," *Procedia Computer Science*, vol. 79, pp. 328-336, 2016. Available: 10.1016/j.procs.2016.03.043.

- [15] G. Dieter and L. Schmidt, Engineering Design, New York: McGraw-Hill, 2009.
- [16] Patel, S., 2014. *Developing Responsive Web Applications With AJAX And JQuery*.  
Birmingham, U.K.: Packt Pub.
- [17] Amazon.com. n.d. [online] Available at: <https://www.amazon.com/Raspberry-Pi-MS-004-00000024-Model-Board/dp/B01LPLPBS8> [Accessed 30 March 2020].
- [18] Amazon.com. n.d. [online] Available at: <https://www.amazon.com/Elecrow-800x480-Interface-Supports-Raspberry/dp/B013JECYF2> [Accessed 30 March 2020].
- [19] Dwyer, G., Aggarwal, S., and Stouffer, J., 2017. *Flask: building python web services*.  
Birmingham, United Kingdom: PACKT Publishing Limited.

## Appendix

### A: Hardware Devices used in the development of the Biometric System

#### *Appendix A: Hardware devices*

Raspberry pi 3B+



5" 800x480 TFT HDMI



Fingerprint sensor R307



PL2303 USB TTL Cable FTDI



**B: Flask server running as localhost to the raspberry pi.**

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 171-142-059
```

*Figure 23: Flask web server running on Raspberry pi*

### C: Folder tree of the system

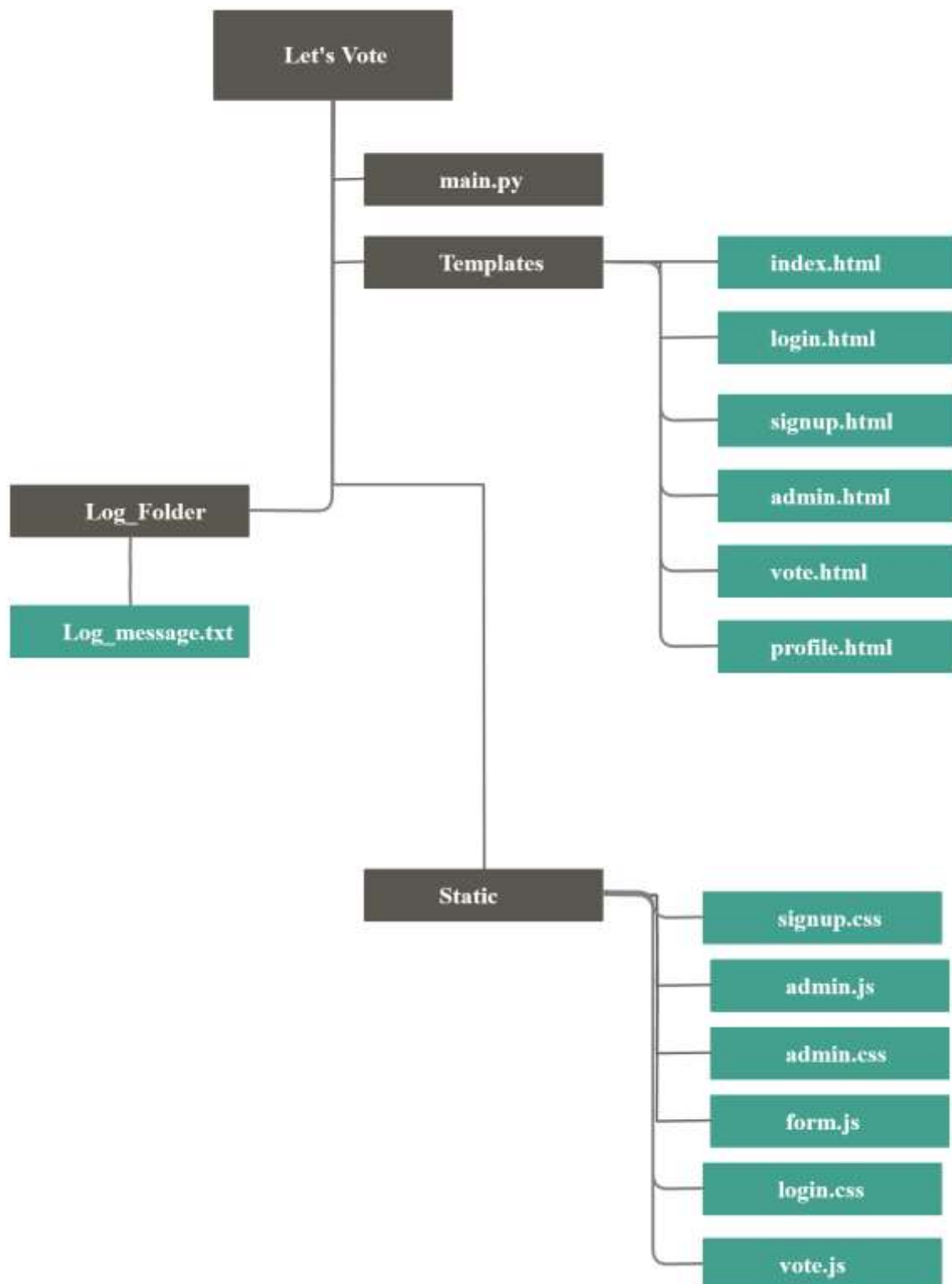


Figure 24: Directory of the project

## D: Flask extensions being imported in the project



```

# importing flask, also importing template to switch within pagesmysql, importing flash, session
from flask import Flask, render_template, \
    flash, session, make_response, request
# importing bootstrap to create nice css interface
from flask_bootstrap import Bootstrap
# importing wtforms to handle signup and login forms easier
from flask_wtf import FlaskForm
# various fields to be used in the application
from wtforms import StringField, PasswordField, BooleanField, TextField, RadioField, Form
# to validate various fields in the form
from wtforms.validators import InputRequired, Length
# importing database using flask-SQLAlchemy
from flask_sqlalchemy import SQLAlchemy
# using datetime to get date for database
from datetime import datetime
# importing redirect and url_for and jquery object
from flask import redirect, url_for, jsonify, request
import json

# for improved security the following were imported
from werkzeug.security import generate_password_hash, check_password_hash

# more flask utility to aid log in process
from flask_login import LoginManager, UserMixin, \
    login_user, login_required, logout_user, current_user
# importing library for admin page
# from flask_admin import Admin, AdminIndexView

# importing pyFingerprint library
from pyfingerprint.pyfingerprint import PyFingerprint
# importing time, lcd driver
import time
# import lcd driver

```

Figure 25: Image of various flask extensions and frameworks imported

## E: Code implementation of the voting system

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error=None
    form =LoginForm()

    #if submission succesfull
    if form.validate_on_submit():
        #call fingerscan route
        result=fingersearch()

        if result is None:
            error= 'Invalid Username, Password or FingerprintID Please try again.'
        else:
            #get hashed prints form the result
            prints= result[1]
            ScanHashed_prints= prints
            #querying database for the user
            user=User.query.filter_by(Username=form.username.data).first() #only use first name for checking
            print(user.id)
            admin=User.query.filter_by(Username='admin').first() #only use for checking admin
            user_json = json.dumps(user.id,cls=AlchemyEncoder)
            session["user_json"]=user_json
            print("user_json" in session)
            #if user exist,check password, and fingerprint
            if admin:
                if check_password_hash(user.Password,form.password.data) and (user.Username== 'admin'):
                    login_user(admin,remember=form.remember.data)
                    return redirect(url_for('admin'))
```

Figure 26: Code implementation of the login system

```
user_json = json.dumps(user.id,cls=AlchemyEncoder)
session["user_json"]=user_json
print("user_json" in session)
#if user exist,check password, and fingerprint
if admin:
    if check_password_hash(user.Password,form.password.data) and (user.Username== 'admin'):
        login_user(admin,remember=form.remember.data)
        return redirect(url_for('admin'))

if user:
    if check_password_hash(user.Password,form.password.data) and (user.Fingerprint== ScanHashed_prints):
        login_user(user,remember=form.remember.data)
        return redirect(url_for('votingpoll'))

error = 'Invalid Username, Password or FingerprintID Please try again.'
#if username/password is invalid
file = os.path.join(
    app.config['LOG_FOLDER'], session.get('log_file')
)

filepath = os.path.join('LOG_FOLDER', 'log_messages.txt')
with open(filepath, 'w') as f:
    pass

filepath = os.path.join('LOG_FOLDER', 'log_messages.txt')
with open(filepath, 'w') as f:
    pass

return render_template('login.html', form = form)
```

Figure 27: Code implementation of the login system (continues)



```

726 #for sign up page
727 @app.route('/signup', methods=['GET', 'POST'])
728 def signup():
729     error=None
730     form = Register()
731
732 #if user wants to submit the form...
733 if form.validate_on_submit():
734     session['Username'] = form.username.data
735     result=fingerscan()
736
737     if not result:
738         error= 'Fingerprint ID Error, Please try Again'
739
740         file= os.path.join(
741             app.config['LOG_FOLDER'], session.get('log_file')
742         )
743
744         filepath = os.path.join('LOG_FOLDER', 'log_messages.txt')
745         with open(filepath, 'w') as f:
746             pass
747
748     else:
749         hashed_prints=result[1]
750         #generate hash password
751         hashed_password = generate_password_hash(form.password.data, method='sha256')
752         #passing parameters of User to new_user to be pushed into the database
753         # global table_prints
754         table_prints= hashed_prints
755         new_user=User(

```

Figure 28: Code implementation of the signup system

```

748     else:
749         hashed_prints=result[1]
750         #generate hash password
751         hashed_password = generate_password_hash(form.password.data, method='sha256')
752         #passing parameters of User to new_user to be pushed into the database
753         # global table_prints
754         table_prints= hashed_prints
755         new_user=User(
756             Username=form.username.data,
757             Password=hashed_password,
758             Date=form.created,
759             Fingerprint=hashed_prints
760         )
761         db.session.add(new_user)
762         db.session.commit()
763         file= os.path.join(
764             app.config['LOG_FOLDER'], session.get('log_file')
765         )
766
767         filepath = os.path.join('LOG_FOLDER', 'log_messages.txt')
768         with open(filepath, 'w') as f:
769             pass
770
771         return redirect(url_for('dashbbard'))
772
773
774     filepath = os.path.join('LOG_FOLDER', 'log_messages.txt')
775     with open(filepath, 'w') as f:
776         pass
777
778     return render_template('signup.html', error=error, form=form)

```

Figure 29: Code implementation of signup System(continues)



```

435 @app.route('/create', methods=['GET', 'POST'])
436 def create():
437
438     form = CreatePollForms()
439     ques = Question.query.filter_by(question=form.question.data).first()
440     context = {'form': form}
441     if form.validate_on_submit():
442         ques = Question.query.filter_by(question=form.question.data).first()
443
444         if ques is None:
445             new_ques=Question(question=form.question.data)
446             db.session.add(new_ques)
447             db.session.commit()
448             user_jso = json.dumps(new_ques.questionid, cls=AlchemyEncoder)
449             session["user_jso"]=user_jso
450
451             return redirect('add')
452
453
454
455
456     return render_template('create.html', ques=ques, form=form)
457

```

*Figure 30 Code Implementation of creating a new poll (Backend script)*

```

496
497 @app.route('/add', methods=['GET', 'POST'])
498 def add():
499     if request.method == 'POST':
500
501         ques = Question.query.filter_by(questionid= user_jso).first()
502
503
504         return render_template('candidate.html', ques=ques)
505

```

*Figure 31: Code Implementation of adding a new candidate to the poll created (Backend script)*

```

1  {{% extends "bootstrap/base.html" %}}
2
3
4  {% block title %}Candidate Poll{% endblock %}
5
6  {% block content %}
7    <div class="row">
8      <div class="col-lg-8 col-lg-offset-2">
9        <div id="memberSection" class="panel panel-default">
10          <div class="panel-heading">
11            <h3 class="panel-title">Enter Candidate Name For: <span id="memberNumber{{ ques.question }}">{{ ques.question }}
12          </div>
13          <div class="panel-body">
14            <div class="form-inline">
15              <div class="form-group">
16                <meta name="csrf-token" content="{{ csrf_token }}">
17                <label for="nameInput{{ ques.questionid }}">Name</label>
18                <input type="text" class="form-control" id="nameInput" required="required">
19              </div>
20            </div>
21          </div>

```

Figure 32: HTML implementation of adding a new candidate to the poll created (Frontend script)

```

17      <meta name="csrf-token" content="{{ csrf_token }}">
18      <label for="nameInput{{ ques.questionid }}">Name</label>
19      <input type="text" class="form-control" id="nameInput" required="required">
20    </div>
21  </div>
22  <button class="btn btn-primary updateButton" me="one">Add Candidate</button>
23  </div>
24  <div class="row">
25    <hr />
26    <div class="col-lg-4">
27      <button class="btn btn-danger doneButton" me="two">Done Adding</button>
28    </div>
29  </div>
30  <div>
31    <div id="successAlert" class="alert alert-success" role="alert" style="display:none;"></div>
32    <div id="errorAlert" class="alert alert-danger" role="alert" style="display:none;"></div>
33  </div>
34
35  <script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.18.2/axios.js"></script>
36  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
37  <script type="text/javascript" src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
38  <script type="text/javascript" src="{{ url_for('static', filename='ajax.js') }}" defer></script>
39
40  {% endblock %}

```

Figure 33: HTML implementation of adding a new candidate to the poll created (Frontend script) continues

```

1
2 $(document).ready(function() {
3
4     $('#updateButton').on('click', function(event) {
5
6         var name = $('#nameInput').prop("required", true).val();
7         console.log(name);
8
9         var csrftoken = $('meta[name=csrf-token]').attr('content')
10
11         $.ajaxSetup({
12             beforeSend: function(xhr, settings) {
13                 if (!/^(GET|HEAD|OPTIONS|TRACE)$/i.test(settings.type)) {
14                     xhr.setRequestHeader("X-CSRFToken", csrftoken)
15                 }
16             }
17         })
18
19
20         req = $.ajax({
21             url: '/update',
22             type: 'POST',
23             data: { name : name}
24
25
26
27         })
28         .done(function(data){
29             if(data.error) {
30                 $('#errorAlert').text(data.error).show();
31                 $('#errorAlert').text(data.error).hide(5000);
32                 $('#successAlert').hide();
33

```

Figure 34: JavaScript implementation of adding a new candidate to the poll created (Frontend script)

```

34
35         })
36         .done(function(data){
37             if(data.error) {
38                 $('#errorAlert').text(data.error).show();
39                 $('#errorAlert').text(data.error).hide(5000);
40                 $('#successAlert').hide();
41             }
42             else{
43                 $('#successAlert').text(data.name).show();
44                 $('#successAlert').text(data.name).hide(3500);
45                 $('#errorAlert').hide();
46                 $('#nameInput').val('');
47             }
48         })
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
```

```

459
460 @app.route('/update', methods=['POST'])
461 def update():
462     name = request.form['name']
463     can = Can.query.filter_by(name=name).first()
464     user_jso=session["user_jso"]
465     print(name)
466     print(user_jso)
467     if name:
468         if can is None:
469             new_can=Can(questionid=user_jso,name=name,vote=0)
470             db.session.add(new_can)
471             db.session.commit()
472
473             return jsonify({'name':'Candidate added! Succesfully'})
474
475
476     return jsonify({'error': 'Please Enter Candidate Name!'})
477

```

Figure 36: Update route gets information posted from add page and adds it to the database

```

494 @app.route('/voting/<poll_id>', methods=['GET', 'POST'])
495 @login_required
496 def voting(poll_id):
497     form = VotingForms()
498     context = {'form':form}
499     poll = Can.query.filter_by(questionid=poll_id).all()
500     global id_poll
501     id_poll=poll_id
502
503     if request.method == 'POST':
504         selected_option= request.form['poll']
505         global select
506         select = selected_option
507
508         #query for table, check if user has already voted
509         user_print=PrefectPrints.query.filter_by(UserID=fpj).all()
510         name = Can.query.filter_by(name=select).first()
511
512
513
514         if user_print is None:
515             name.vote=name.vote+1
516             db.session.commit()
517
518             new_name = PrefectPrints(questionid=poll_id, UserID=fpj)
519
520             db.session.add(new_name)
521             db.session.commit()
522             return redirect(url_for('result', poll_id=poll_id))
523

```

Figure 37: Code implementation of the voting page (Backend Script)



```

524
525     elif user_print is not None:
526         l= []
527         for i in user_print:
528             l.append(i.questionid)
529
530         print(l)
531
532         print(selected_option)
533         choose=Can.query.filter_by(name=selected_option).first()
534         if (int(poll_id) not in l):
535
536             name.vote=name.vote+1
537             db.session.commit()
538
539             new_name = PrefectPrints(questionid=poll_id,UserID=fp)
540
541
542             db.session.add(new_name)
543             db.session.commit()
544             return redirect(url_for('result',poll_id=poll_id))
545
546         else:
547             return ('Invalid form',404)
548
549         return redirect(url_for('result',poll_id=poll_id))
550     return render_template('vote.html',context=context,form=form)

```

Figure 38: Figure 19 Code implementation of the voting page (Backend Script) continues

```

102 #Creating a table for the user
103 class User(UserMixin, db.Model):
104     __tablename__ = 'user_details'
105     #id column
106     id= db.Column('id', db.Integer,primary_key= True)
107     Username= db.Column('Username',db.Unicode,unique=True)
108     Password= db.Column('Password',db.Unicode)
109     Fingerprint=db.Column('Fingerprint',db.Unicode,unique= True)
110     Date=db.Column('Date',db.Unicode)
111     Prefect= db.relationship('Prefect',backref='User',uselist=False)
112     #EPrefect= db.relationship('EPrefect',backref='User',uselist=False)
113
114
115
116
117 #cretaing table for school prefect position
118 class Prefect(db.Model,SerializerMixin):
119     serialize_rules=()
120     __tablename__ = 'senior_school_prefect'
121     #table columns
122     ID= db.Column('ID', db.Integer,primary_key= True)
123     UserID=db.Column('UserID', db.Integer, db.ForeignKey('user_details.id'))
124     Candidate= db.Column('Name',db.Unicode,unique=True)
125     Votes= db.Column('Votes', db.Integer)
126     # PrefectPrints= db.relationship('PrefectPrints',backref='Prefect',uselist=False)
127

```

Figure 39: Creating database tables as classes in code OSS IDE

```

227
228 #using classes to create the login form
229 class LoginForm(FlaskForm):
230     username = StringField('username',validators=[InputRequired()])
231     password= PasswordField('password',validators=[InputRequired(),Length(min=6,max=80)])
232     remember = BooleanField('remember me')
233
234
235 #using classes to create the profile form
236 class ProfileForm(FlaskForm):
237     firstname = StringField('Firstname',validators=[InputRequired()])
238     surname = StringField('Surname',validators=[InputRequired()])
239
240
241 #Class for registration
242 class Register(FlaskForm):
243     username = StringField('username',validators=[InputRequired()])
244     password= PasswordField('password',validators=[InputRequired(),Length(min=6,max=80)])
245     created= datetime.utcnow()
246

```

Figure 40: Creating Flask Forms as class